Scalable Flow based Management Scheme in Software Define Network (SDN) using sFlow

ADENIJI OLUWASHOLA DAVID[1], OLUWABUSAYO ISRAEL OMOTOSHO[2*]
[1]Computer Science Department,
University of Ibadan,
NIGERIA

[2]Computer Science and Engineering Department,
Santa Clara University, California,
USA

[*]*Corresponding Author*

Abstract: - The threats to information privacy while connected to cyber space are capacious and complex which require resilient network and antifragile security mechanisms. Software Define Network (SDN) infrastructure itself is predisposed to severe threats that may damage the provision of its usability as a security provider. The essential qualities of (SDN) are to provide support for high bandwidth and timely content delivery. SDN granular approach to security by centralizing the security control into one entity using the controller to ensure service control and information protection. SDN provides a new paradigm for applications to interact with the network. This interaction with declarative abstraction will instruct the Application Programming Interface (APIs) to direct the configuration and operation of the network. The API is queried to ask the network for information in order to plan and optimize the network operations. In this study, the vulnerability exploited by attackers to perform distributed denial of service (DDoS) attacks is examined. The trust between the control planes and forwarding planes is crucial in SDN. The separation of the control and data planes contributes to open security challenges such as denial of service (DoS) attacks, man-in-the-middle attacks, and network saturation attacks. The platform runs on Mininet 2.2.2, Ubuntu 18.04, Ryu Controller 4.34, and Sflow-RT. The Classification learning is based on Support Vector Machine (SVM). The contribution is to provide monitoring application of Flow RT Status and SFlow RT Packet Monitoring during Normal Traffic Generation. The implication for the monitoring application of SFlow RT Status is to supervise the failure in the status of sFlowAgent, sFlow Byte, and sFlow packet against cyber-attack.

Key-Words: Software Define Network, Cyber Threat, DDos, SFlow

Received: April 9, 2022. Revised: April 17, 2023. Accepted: May 11, 2023. Published: June 22, 2023.

1 Introduction

Software-defined networking (SDN) is radically changing the design of network architect-Ture. The separation of control and data plane in SDN ensures the program controls logic and instructs the data to be forwarded accordingly. The DDoS attacker relies on sending an irresistible number of fake packets thereby reducing the performance of resources such as CPU, memory, and network bandwidth. SDN Flow statistics enable network providers to manage their network resources more efficiently by having a clear overview of network resource utilization. Packet sampling is crucial factors that affect the controller's workload and bandwidth consumption. sFlow can associate with both hardware and software switches so that sFlow can collect data traffic from switches' interfaces. Also, Flow samples collected by embedded sFlow agent in switch or router will be compiled for traffic data analysis. Furthermore, sFlow datagrams consist of interface samples that are collected from the switch's ports that performed packet sampling function. Open Flow controllers experience overhead issues and several researchers have proposed new architectures to resolve the challenges. Open Flow used a pull-based mechanism as their read-state, which causes interference with flow entries and consumes bandwidth during the data gathering state. Aggregating counters via wildcard Techniques will destabilize the controller's ability to manage specific elephant flows. As a result, all flows

regardless of size will be forwarded to the controller. In view of this, a controller like Floodlight will not only utilizes available link bandwidth but focus on finding the optimal path. It is possible that flows might share the same path instead of being routed to an alternate path. The implication is that link bandwidth consumption on the current path will increase.

2 Problem Formulation

The centralized control and programmability of SDN introduce new faults when an attack is observed in the network, [1], [2], which result in new threats that are unbreakable to avoid. To solve the problem of attacks there is a need to implement techniques that will mitigate resource wastage as proposed in [3], [4]. The flooded-based Distributed Denial of Service (DDoS) attacks are the most advanced threat challenges to network security, [5], it is desirable to design lightweight and scalable DDoS mitigation methods. There are a lot of SDN security mechanisms to address DDoS attacks. These new techniques in SDN provide a solution to DDoS attacks in traditional networks which ranges from detection of DDoS, [6], [7], to DDoS defense, [8]. Development of DDoS Attack Detection Approach in Software Defined Network Using Support Vector Machine Classifier in [9]. The security Techniques will protect it from any sign of vulnerabilities associated with the network such as distributed denial of service (DDoS) attacks, [10]. The review in [11], the focus of the study is to develop an IPv6 packet matching mechanism in OpenFlow Software Defined Network using Flow Label. The prime focus of this study is to develop a mechanism for network function virtualization in an IPV6-enabled SDN. The analysis of the percentage line rate for IPv4 was 95.27% as compared to the developed model of IPv6 with a line rate of 54.55% for each network slice, [12].

3 Problem Solution

The developed flow-based management scheme is emulated using Mininet with a Network topology of ten (10) Hosts and one (1) switch. The sFlow Management scheme for the developed model is presented in Fig 1 below.

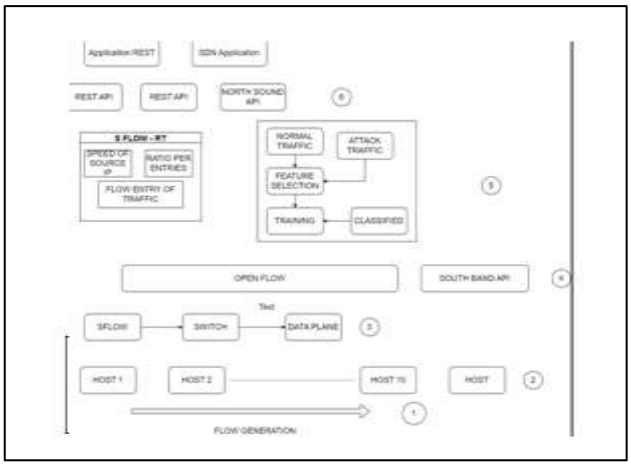

Fig. 1: sFlow Management scheme

There are six stages involved which are specifically: flow generation, end-to-end transmission of the switches, sFlow data communication, sFlow data collection and detection, Flood light controller, and classification.

1. Stage 1: Flow Generation: Wireshark was used as an open-source program for capturing packets and logging traffic over the network. The flow of traffic was generated via iPerf, iperf output contains a time-stamped report of the amount of data transferred and the throughput measured.
2. Stage 2: End-to-end transmission of the switches: the flows are transmitted from a host from end to end via the switch, the switch with embedded sFlow agent will collect traffic samples and compile them in datagram format.
3. Stage 3: sFlow Data Communication consist of sFlow agent traffic which includes the application, scripting, and HTTP Connection. In this stage, the sFlow agent keeps track of the sFlow byte and sFlow packet.
4. Stage 4: sFlow data collection and detection: The Open Flow protocol is programmed to design the network protocol and direct traffic among routers and switches. Using a mininet network simulator, the network topology has 10 hosts/nodes and one (1) single open flow switch connected to one ryu controller.
5. Stage 5: Controller Plane: Floodlight controller was used to perform simple forwarding. The forwarding flow is the Normal traffic and attack traffic. The study lunch the DDos attack traffic and the normal traffic. Feature selection of both the

normal and attack traffic was trained in the model.

6. Stage 6: Classification: A Support Vector Machine was used as the algorithm for classification. sFlows communication is filtered according to IP source, destination address, input port, output port, and flow direction. The sFlow's REST API enables metrics-based customization to filter inbound or outbound flows.
7. The controller can re-route flows to alternate paths via reactive routing. The reason for re-routing flows is to alleviate link bandwidth between switches in the default path. The Ryu Controller during configuration starts the simulation at port 6653 and Mininet creates one virtual switch and 10 hosts, a ping is run to generate normal traffic for 20 minutes as presented in Fig 2 below.

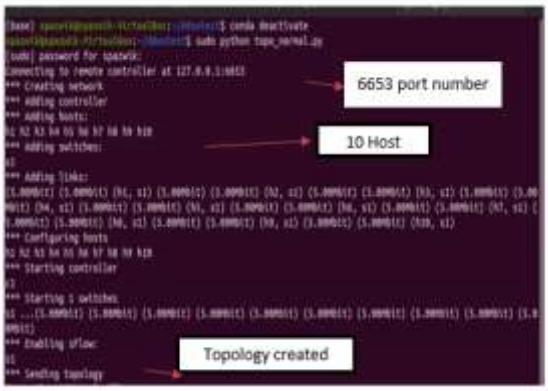

Fig. 2: Configuration of SDN Network Topology

Two traffic generation was obtained in this section of the implementation. The Normal Traffic and the DDos attack generation are embedded sFlow agents. Fig 3 below presents the normal traffic generation.

Fig. 3: Generating Normal Traffic SDN

The data plane switches are to assign output ports by pairing for different flows to reduce the controller's overhead. In this manner the number of a packet sent into the controller is minimized, however, the request for new flow entries are required during pairing. The attack traffic log is shown below in Fig 4.

Fig. 4: Generating Attack Traffic SDN

4 Results and Discussion

This section presents the various results during the implementation which are based on prediction of normal and attack traffic generation. The prediction of networks is basically to disintegrate knowledge due to hardware and software configuration. Configuration tests were conducted on the SDN platform. Normal and attack traffic was sent to the network from different ports. The Normal **Traffic Log of SFlow RT monitoring Status** by the RYU Controller created 500+ samples of datasets for normal traffic. The test1 for the monitoring log was presented in Fig. 5.

Fig 5: SFlow RT Status during Normal Traffic Generation

Fig. 5 shows the first SFlow Agents Test with 11 Apps, 11 scripts, and 1 HTTP connection, SFlow Bytes 2.86 Kbps, and 0.79 pps sFlow packets. In another status monitoring Fig 6 below shows the second test, the SFlow Agents have 11 Apps, 11 scripts, and 1 HTTP connection, SFlow Bytes 17.1 Kbps, and 2.19 pps sFlow packets. The sFlow agent on the switch provides ingress information of all forwarded layer 2 and layer 3 traffic on LAG and

Ethernet ports. The tests are conducted for 1200 seconds with 2 seconds intervals for traffic collection on normal traffic and attack traffic. *The Attack Traffic Log by the RYU Controller created* 600+ samples of attack traffic. A similar test 2 was conducted as presented in the traffic generation. Investigating the result in the attack traffic below, there was zero (0) failure in the App, zero (0) failure in the script, and zero (0) failure in the HTTP connection of the agent The test2 monitoring attack traffic is presented in the Fig 6 below.

Fig. 6: SFlow RT Status during Attack Traffic Generation

The cognitive behavior of the developed Application provides the zero failure connection of the agent as the result of transited packet headers by sFlow agents on transit. The information deduces shows that the inner and outer source with destination addresses can be decoded while the tunneling protocol does not encrypt the inner header. SFlow RT Packet Monitoring during Attack Traffic Generation is presented below. This experiment uses the Ping Flood attack to elaborate the execution of the developed system. The h2 with Ping Flood 150,000 packets per second traffic rate shows that ping flood attack generates around 100,000 packets per second traffic rate as shown in Fig. 7.

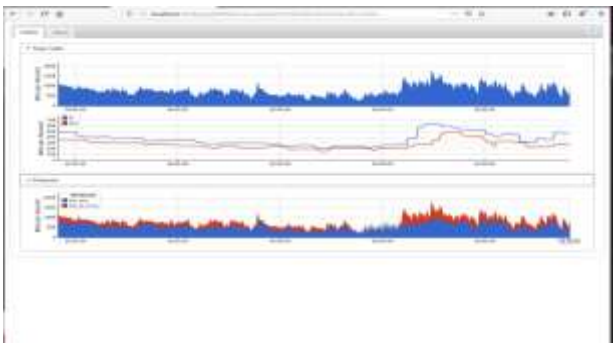

Fig. 7: SFlow RT Packet Test 1 during Attack Traffic Generation

The developed monitoring application identifies the three most famous DDoS attacks; Ping Flood, Ping of Death, and SYN Flood in real-time with performance varying from 80,000-130,000 packets per second. Flow Entry of Traffic is how network traffic coming into the network has a number of flow counts. Normal traffic will have fewer flow counts than during a DDOS attack traffic as presented in Fig 8 below.

Fig. 8: SFlow RT Packet Test2 during Attack Traffic Generation

It was observed that TCP flow after sFlow-RT was detected at switch-1 with port s1 eth10. The flow detected on the switch--1 exceeded the flow threshold, consuming high bandwidth between switch-1 and hosts thereby utilizing all available network resources. However, the controller does not account for bandwidth competition between flows which has resulted in link congestion between switch-1 and switch-3 even when an alternate path is available. Furthermore, the link between switch-1 and switch-2 was underutilized. However, flow mitigation is expected to reduce bandwidth competition between flows that were sharing the same path and conserve the processing power of switches as shown in Fig. 9.

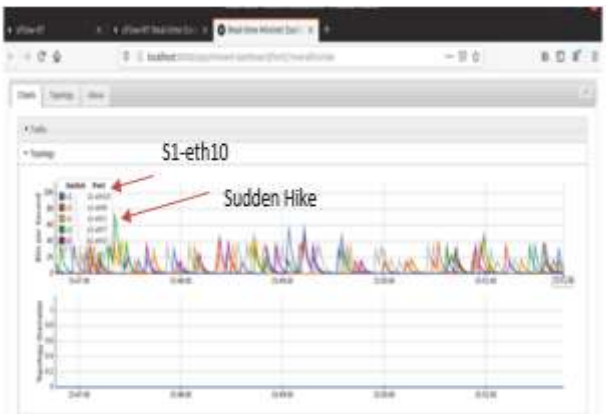

Fig. 9: flow mitigation of TCP traffic

After mitigation, it is expected that subsequent flows will be lesser than the flow threshold. The default path has been switched to an alternate path that includes switch-1-ethe1, switch-2-eth2, and switch-4-eth4 as shown below in Fig 10.

Fig. 10: flow After mitigation of TCP traffic

The below script in Fig 11 was used to calculate the detection ratio, false detection rate, and false positives during prediction. The detection rate was 98% with a false positive of zero.

Fig. 11: Ratio of False Positive on Jupyter Notebook

The controller needs to have sufficient data of both normal and attack traffic for the machine learning algorithm to predict the attacks. For normal attacks, the SSIP is usually low and for attacks, the count is usually higher. Fig 12 below presents the result. **Speed of Source IP:** This feature is the total number of Traffic Packets coming into the network from the source IP within a particular time interval given below.

$$SSIP = \frac{SumIPsource}{T}$$

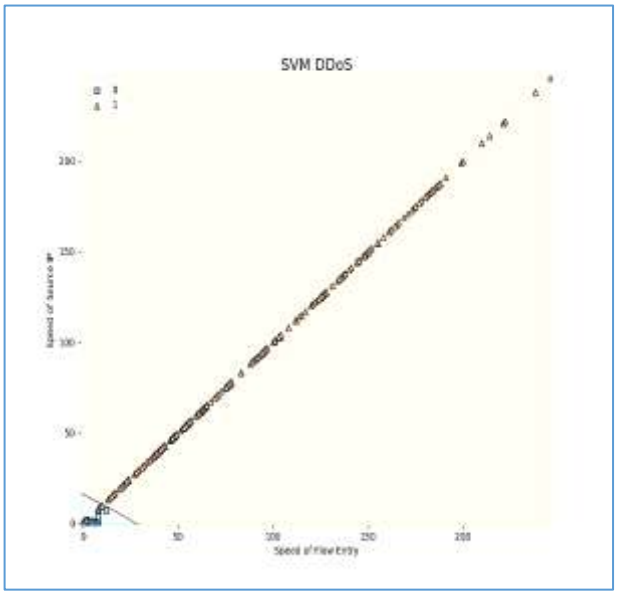

Fig. 12: SVM Graph for SSIP and SFE

4 Conclusion

The collection of data increases due to network expansion, SDN's central control of the network for knowledge was to learn and offer processes to support cognitive behavior. The SFlow RT Status during Normal Traffic Generation provides real-time monitoring for pipeline measurement, flow collection, and analysis of the programmable stages as conducted in the network topology. It was observed during the implementation that the resilience of distributed messages can be improved.

Acknowledgments:
The authors wish to thank the Department of Computer Science, University of Ibadan, Nigeria for the support in this research work.

References:

[1] E. B. H. Tan and Y. W. Chong, "An optimized flow management mechanism in OpenFlow network," in 2017 Int. Conf. on Information Networking, Da Nang, Vietnam, IEEE, pp. 143–147, 2017.

[2] M. H. Abidi, H. Alkhalefah, K. Moiduddin, M. Alazab, M. K. Mohammed et al., "Optimal 5G network slicing using machine learning and deep learning concepts," Computer

Standards & Interfaces, vol. 76, no. 1, pp. 103518, 2021.

[3] M. H. Abidi, H. Alkhalefah, K. Moiduddin, M. Alazab, M. K. Mohammed et al., "Optimal 5G network slicing using machine learning and deep learning concepts," Computer Standards & Interfaces, vol. 76, no. 1, pp. 103518, 2021.

[4] J. R. Correa and M. X. Goemans, "Improved bounds on nonblocking 3-stage clos network," SIAM Journal on Computing, vol. 37, no. 3, pp. 870–894, 2007.

[5] R. H. Jhaveri, R. Tan and S. V. Ramani, "Real-time QoS-aware routing scheme in SDN-based robotic cyber-physical systems," in IEEE 5th Int. Conf. on Mechatronics System and Robots, Singapore, pp. 18–23, 2019.

[6] S. Y. Hassas and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in Proc. of the First Workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, pp. 19–24, 2012.

[7] A. Tootoonchian and Y. Ganjali, "Hyperflow: A distributed control plane for openflow," in Proc. of the 2010 Internet Network Management Conf. on Research on Enterprise Networking, San Jose, USA, vol. 3, 2010.

[8] X. Nguyen, D. Saucez, C. Barakat and T. Turletti, "Rules placement problem in OpenFlow networks: A survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1273–1286, 2015.

[9] Adeniji, O.D., Adekeye, D.B., Ajagbe, S.A., Adesina, A.O., Oguns, Y.J., Oladipupo, M.A. *Development of DDoS Attack Detection Approach in Software Defined Network Using Support Vector Machine Classifier*. In: Ranganathan, G., Bestak, R., Fernando, X. (eds) Pervasive Computing and Social Networking. Lecture Notes in Networks and Systems, vol 475.pp319-331, 2022.

[10] O. Ashimi Quadri and Adeniji Oluwashola David. Detection and Mitigation of Flood Attacks in IPv6 Enabled Software Defined Networks. Advances in Research, 21(8): 1-9, Article no.AIR.57485. 2020.

[11] A. A. Olabisi, O. D. Adeniji , Abeng Enangha. A Comparative Analysis of Latency, Jitter and Bandwidth of IPv6 Packets Using Flow Labels in Open Flow Switch in Software Defined Network. Vol. 1, Issue 3, July 2019, pp. 30 – 36, 2019 Afr. J. MIS. https://afrjmis.net.

[12] Adeniji, O.D., Ayomıde, M.O., Ajagbe, S.A. *A Model for Network Virtualization with OpenFlow Protocol in Software-Defined Network*. In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2022.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflict of interest to declare.